High Level Architecture (HLA) Performance Framework

Dr. Russ Richardson
Dr. Judith Dahmann
Dr. Richard Weatherly
Mr. Richard Briggs

ITEC 28 April 1998

Topics

- Motivation and Objectives
- Federation Execution Planners Workbook
- Performance Benchmark Programs
- Supporting Tools
- Conclusion

What is the Performance Framework?

- A Common Framework for defining the aspects of HLA federations which bear on their runtime performance
- Foundation for planning federation executions and defining performance of HLA federation components
  - -Run-time Infrastructure (RTI)
  - -Federate Capabilities
  - -Hardware Requirements
  - -Network Requirements
- Comprised of two basic components
  - -Federation Execution Planners Workbook
  - -Performance Benchmark Programs

Purpose of the Federation Execution Planner's Workbook (FEPW)

- **Initially to understand RTI performance requirements and capabilities**
  - How HLA is being used
  - Performance needs of actual federations
  - Input for benchmark definitions for RTIs
- **Now supporting broader uses**
  - Planning of federation executions
    - Insight into the entire planning process
    - Understanding relationships among federates
    - Understanding timing and coordination requirements
  - Documenting prior federation executions
  - Functional prototype for FEPW tool

What is The FEWB

Set of Excel tables forming an Excel Workbook

- **Summary Tables**
- **Host Table**
- **LAN Tables**
- **RTI Services Tables**
- **Object/Interaction Tables**
- **Cross-Reference Tables**
- **Data Representation Tables**

Summary Tables

- **Information about the federation execution**
  - **Name**
  - **Number of concurrent federation executions**
  - **RTI used**
- **Summary information about each federate**
  - **API used**
  - **Size**
  - **Tick and timing data**
- **Identifies hosts and LANs used by the federates**

Federation Execution Summary Table

Federation Execution Name		Sample
Concurrent Federation Executions	Number	4
	Names	Test
		Demo
		Prototype
RTI Software Used	Vendor	DMSO
	Version	1.3

Federate Summary Table

Fed	Name	API	Size (MB)	Tick			Time Manageme	
				Rate (/sec)	Min Value	Max Value	Regulating (y or n)	Constrained (y or n)
1	Land_Simulation	Ada	10	10	0.001	0.005	Y	Y
2	Land_Simulation	Ada	15	10	0.001	0.005	Y	Y
3	Air_Simulation	C++	8	50	0.001	0.005	Y	Y
4	Sea_Simulation	Java	7	20	0.001	0.005	Y	Y
5	Data_Logger	Java	1	20	0.001	0.005	N	N
6	Viewer	C++	1	10	0.001	0.005	N	Y

Host Table

- One per federation execution
- Information on hardware
  - Computer make and model
  - Operating system
- Memory capacity of the computer
- Execution time available on the computer

Host Table

NOTE: One table per Federation

Host	Computer			Operating System		Memory (MB)		% CPU Available to RTI	Notes
	Vendor/ model	# CPUs	Speed (MHz)	Name	Version	Total (MB)	Available (MB)		
1	Sun Ultra 2	1	300	Solaris		192			
2	Sun Ultra 2	1	200	Solaris		192			
3	SGI Octane	1	300	IRIX		256			
4	Dell latitude CP	1	233	NT		64			
5	Sun Ultra 2	1	250	Solaris		192			
6	Compaq	1	233	NT		128			
7									
8									
9									
10									

LAN Tables

- One per federation execution
- Descriptive information on each LAN used
  - Type of network
  - Throughput
- LAN-to-LAN connections
  - Device
  - Throughput
  - Latency

NOTE: One table per Federation

LAN Description Table

LAN	Physical Type (Ethernet, ATM,etc.)	Bandwidth	
		Specified	Available
1	Ethernet		
2	Ethernet		
3			
4			
5			
6			

LAN-to-LAN Connectivity Table

LAN	1	2	3	4	5
1					
2	Device Bandwidth Latency				
3	Device Bandwidth Latency	Device Bandwidth Latency			
4	Device Bandwidth Latency	Device Bandwidth Latency	Device Bandwidth Latency		
5	Device Bandwidth Latency	Device Bandwidth Latency	Device Bandwidth Latency	Device Bandwidth Latency	
6	Device Bandwidth Latency	Device Bandwidth Latency	Device Bandwidth Latency	Device Bandwidth Latency	Device Bandwidth Latency

RTI Services Table

- **One per *federate* and *summary for federation***
- **Lists current suite of RTI services**
- **Indicates services used at least once**

RTI Services Table

NOTE: One table per Federate

Federate	1	Land_Simulation

	Service	IF Spec Ref	Used ?		Service	IF Spec Ref	Used ?
Federation	Create Federation Execution	4.2	Y	Ownership M	Unconditional Attribute Ownership Divestiture	7.2	
	Destroy Federation Execution	4.3			Negotiated Attribute Ownership Divestiture	7.3	
	Join Federation Execution	4.4	Y		Request Attribute Ownership Assumption †	7.4	
	Resign Federation Execution	4.5	Y		Attribute Ownership Divestiture Notification †	7.5	
	Register Federation Synchronization Point	4.6			Attribute Ownership Acquisition Notification †	7.6	
	Confirm Synchronization Point Registration †	4.7			Attribute Ownership Acquisition	7.7	
	Announce Synchronization Point †	4.8			Attribute Ownership Acquisition If Available	7.8	
	Synchronization Point Achieved	4.9			Attribute Ownership Unavailable †	7.9	
	Federation Synchronized †	4.10			Request Attribute Ownership Release †	7.10	
	Request Federation Save	4.11			Attribute Ownership Release Response	7.11	
	Initiate Federate Save †	4.12			Cancel Negotiated Attribute Ownership Divestiture	7.12	
	Federate Save Begun	4.13			Cancel Attribute Ownership Acquisition	7.13	

Object/Interaction Tables

- One per *federate*
- Attributes updated by the federate
  - How often
  - In what groupings
- Attributes to which the federate subscribes
  - Latency constraints on updates
- Interactions generated by the federate
  - How often
- Interactions to which the federate subscribes
  - Latency constraints

Federate	1	Land_Simulation

Interaction Table

Interaction Class	Parameter	Size (bytes)	Send					
			y/n		Nominal Rate	Maximum Rate	Condi-tions	Group-ing
Interaction1								
	Parameter1	4	y					A
	Parameter2	4	y					B
	Parameter3	4	y					A

Object Table

Object Class	Attribute	Size (bytes)	Update									Subscribe		Ownership	
			y/n	Count	Nominal Rate	Maximum Rate	Condi-tions	Group-ing	Transport (reliable or best effort)	Ordering (TSO or RO)	Routing Space	y/n	Max latency (msec)	Transfer Rate	Group-ing
Ground_unit				10											
	Attribute1	4	y					A	R	RO	RO	y	200	1/fedex	A
	Attribute2	4	y					B	R	RO	RO	n			

Grouping Description Table

Type	Grouping	Description
Update	A	Position change
Update	B	Damage to unit

Cross Reference Tables

- One per federation
- Attributes updated or reflected by each federate
- Attributes transferred or accepted by each federate
- Interactions initiated or sensed, or reacted to by each federate

NOTE: One table per Federation

Object Attribute Update/Reflect Table

Object Class	Attribute	Federates									
		1	2	3	4	5	6	7	8	9	10
Ground_unit	Attribute1	U/R	U/R	R	R	R	R				
	Attribute2	U/R	U/R	R	R	R	R				
	Attribute3	U/R	U/R	R	R	R	R				
	Attribute4	U/R	U/R				R				

Object Attribute Transfer/Accept Table

Object Class	Attribute	Federates									
		1	2	3	4	5	6	7	8	9	10
Ground_unit	Attribute1	T/A	T/A								
	Attribute2	T/A	T/A								
	Attribute3	T/A	T/A								
	Attribute4	T/A	T/A								

Interaction Initiate/Receive Table

Interaction Class	Parameter	Federates									
		1	2	3	4	5	6	7	8	9	10
Interaction 1	Parameter1	I	R	R			S				
	Parameter2	I/R	R	R	I/R		S				
	Parameter3	I/R	R	R	I/R		S				
	Parameter4	I/R	R	R	I/R		S				

Data Representation Tables

- One per federation
- How data is represented in the federation
- Byte ordering
- Complex data sequence and packing

NOTE: One table per Federation

Base Type Definition Table

Data Type	Size	Code	Notes
Byte ordering	n/a	BigEndian	Big endian (most significant bit first)
float	32	OMT	IEEE single-precision floating point number
double	64	OMT	IEEE double precision floating point number
short	16	OMT	16-bit two's complement integer value in the range -2^{15} to $2^{15}-1$
unsigned short	16	OMT	16-bit integer value in the range 0 to $2^{16}-1$
long	32	OMT	32-bit two's complement integer value in the range -2^{31} to $2^{31}-1$
unsigned long	32	OMT	32-bit integer value in the range 0 to $2^{32}-1$
long long	64	OMT	64-bit two's complement integer value in the range -2^{63} to $2^{63}-1$
unsigned long long	64	OMT	64-bit integer value in the range 0 to $2^{64}-1$
char	8	OMT	8-bit quantity with a numerical value between 0 and 255
boolean	1	OMT	1 bit quantity which can take only the values 0 or 1
octet	8	OMT	8-bit quantity guaranteed not to undergo any conversion
any	n/a	OMT	permits the specification of values which can express any basetype
string	n/a	OMT	One-dimensional array of chars which is terminated with a null (0) char
sequence	n/a	OMT	One-dimensional array of any base type with max size and length

Complex Data Type Definition Table

Complex Type	Elements			Notes
	Element	Type	Size	
Type 1	Element1	float	32	
	Element2	short	16	
		padding	16	
	Element3	long	32	
	Element4	boolean	1	

Performance Benchmark Programs

- **Goals of the Benchmark Programs**
  - **Performance indicators for each of the major categories of inter-federate exchange through the RTI.**
  - **Simple and unambiguous tools that can be applied by general users of the RTI.**
  - **Easy to understand metrics that facilitate comparison and investigation of factors influencing federation performance.**
  - **Source code that can be easily distributed and compiled on all RTI supported platforms.**
  - **Benchmark programs that are parameterizable using simple command-line arguments and FED file modifications.**

Presently 4 Benchmark Programs Defined

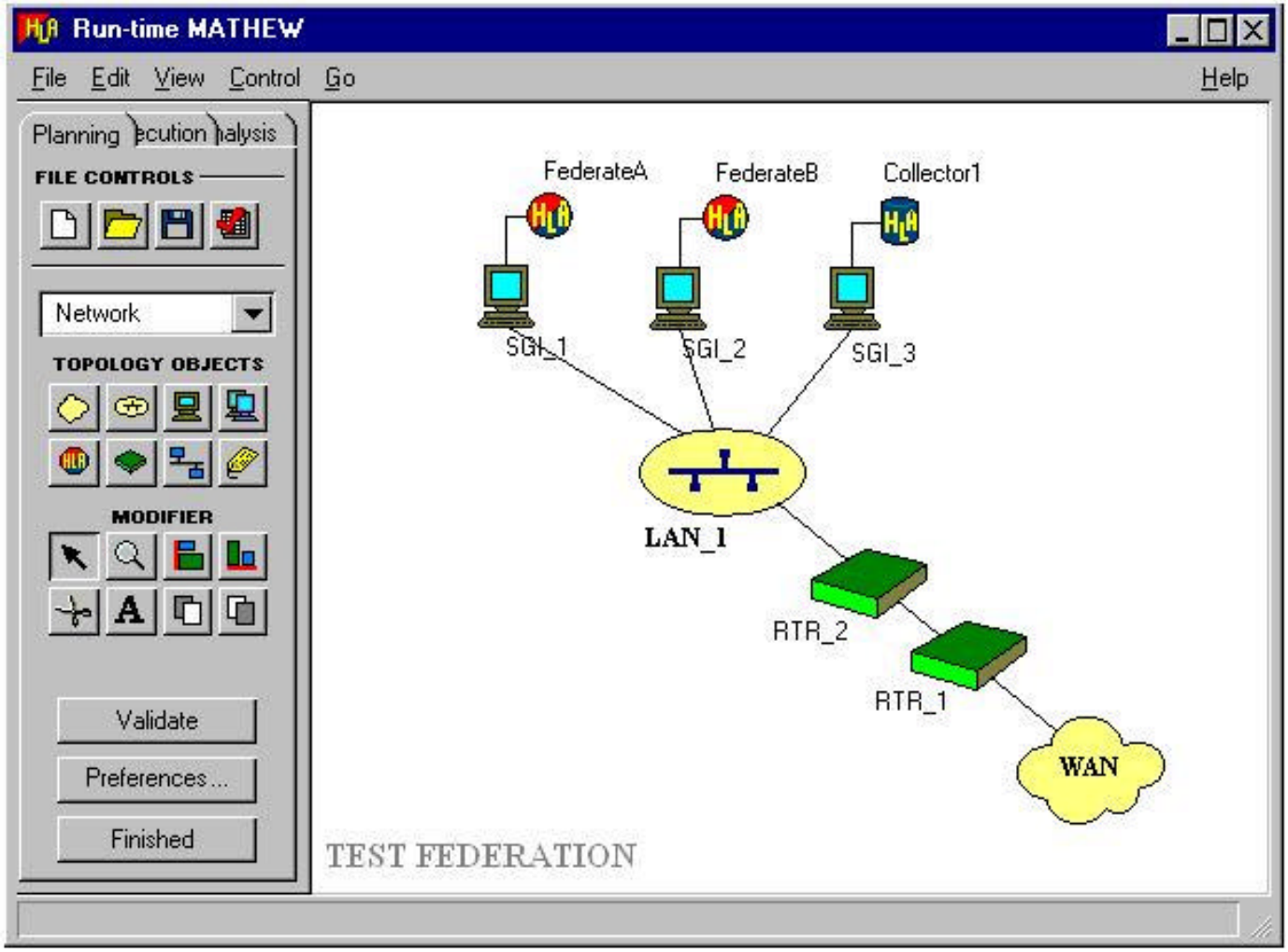
- **Update Latency Benchmark**
  - measures the round trip time(update latency) for an Update Attribute Values(UAV) service call
  - key arguments: size of attribute, number of federates, number of objects per federate
- **Update Throughput Benchmark**
  - measures the number of update attribute values per second that are possible for the given system configuration
  - key arguments: size of attribute, number of federates, number of objects per federate

Presently 4 Benchmark Programs Defined (continued)

- **Time Synchronization Benchmark**
  - measures the number of RTI time step cycles that can be processed by the RTI per second
  - key arguments: number of federates, lookahead for the federates
- **Ownership Management Benchmark**
  - measures the number of round trip ownership transfers per second per federation execution
  - key arguments: number of federates, number of objects per federate
- **In Addition:**
  - Developing standard scripts for running benchmarks

FEPW (FedExec Planner's Workbook) Editor

- **DMSO is building a number of tools that will support the performance framework (e.g. FEPW Editor)**



Conclusion

- **Processes, metrics and tools are needed to support user design of HLA federations to meet specific performance requirements of applications**
- **The performance framework is the first step**
  - **Represents input from variety of users in developing HLA federations**
  - **It provides a structured and consistent way to assists the Federation execution planner**
  - **Provides a common frame of reference for the future development of tool, metrics, and federations**